IN THE CLAIMS

Please amend the claims as follows:

1-2. (canceled)

3. (previously presented) The method of claim 14, wherein storing variations is performed using a pointer.

4. (previously presented) The method of claim 14, further comprising, before presenting the represented sequence to a user:
identifying, replets that can be used to represent multiple subsequences.

5. (currently amended) The method of claim 14, further comprising, before presenting the ~~reconstructed represented~~ first instance of the sequence to a user:
segmenting the matching subsequences into multiple parts to account for location-specific variations of the matching subsequences in the sequence data.

6. (currently amended) The method of claim 14, further comprising, before presenting the ~~reconstructed represented~~ second instance of the sequence to a user:
storing replet information in a replet-information table using a pointer, so that equivalent replet sequences occupy single storage space.

7- 9. (canceled)

10. (currently amended) The method of claim 14, further comprising, before presenting the ~~reconstructed represented~~ second instance of the sequence to a user:
storing multiple views of the sequence data at multiple levels of abstraction.

11-13. (canceled)

14. (currently amended) A computer system-implemented method for storing and presenting sequence data, comprising:

i) specifying a set of replets for analysis by a computer system;

ii) for each replet in the set, comparing each replet by the computer system to a sequence for determining by the computer system a subsequence of the sequence that matches each replet, if any, wherein the sequence represents a genome sequence;

iii) generating, ~~entries of match-set data structures~~ by the computer system responsive to the comparing, ~~one match-set data structure~~ respective entries of a match-set data structure, the match-set data structure having respective entries for the ~~each~~ respective replets, ~~a match-set data structure~~ each entry comprising a sequence identification to identify a ~~sub~~sequence where a match occurs and offset information to determine a position within the sequence where the matching subsequence of the sequence is located, wherein the offset information comprises a first and second position parameter~~, the first position parameter denoting a location in the sequence and the second position parameter denoting an offset from the location~~;

iv) storing the generated entries of the match-set data structures in a computer readable memory by the computer system;

v) deleting by the computer system each matching subsequence from the sequence where it is found;

vi) concatenating by the computer system, in order, unmatched regions of sequences that remain after deleting each matching subsequence to form and store a backbone sequence;

~~vii) the computer system generating a first representation of the sequence responsive to the stored backbone sequence and at least a portion of the stored match-set data and presenting the represented sequence to a user of the computer system;~~

~~wherein one of the subsequences is matched by a certain plurality of the replets and the method further includes:~~

vii) ~~viii)~~ the computer system generating or receiving a selection of at least a first one of the replets, wherein at least a portion of one of the matching subsequences is matched by a certain plurality of the replets and the received selection includes a selection of at least one of the certain plurality of replets;[,]

viii) the computer system generating a first instance of the sequence and presenting the first instance of the sequence to a user of the computer system, wherein the generating of the sequence is responsive to the stored backbone sequence and responsive to the at least one of the stored match-set data entries corresponding to the selected at least first one of the replets, ~~first representation in step vii) is responsive to the selected one of the certain plurality of replets but not responsive to any non-selected replet,~~ wherein the first position parameter of each match-set data structure entry denotes a location in the sequence and the second position parameter of each match-set data structure entry denotes an offset from the location,~~and~~ wherein the selected at least first one of the replets has a position within the sequence, and wherein after receiving the selection the computer system updates the first and second parameters of the entries in the match-set data structure ~~for any non-selected replet associated with the selected one of the replets~~, the updating being responsive to the position of the selected at least first one of the replets;

ix) the computer system generating or receiving a selection of a second one of the replets~~ that was not selected in step viii)~~; and

x) the computer system generating and presenting a second instance ~~representation~~ of the sequence to a user ~~, wherein the representation in step x) is~~ responsive to the at least second one of the replets ~~indicated by the selection in step ix) instead of the one of the replets indicated by the selection in step viii)~~ and wherein the computer system performs the generating of the second instance of the sequence ~~in step x)~~ by reference to the updated first and second parameters for the second one of the replets~~ indicated by the selection in step ix), such that the computer system performs the generating in step x) using the position parameters updated in step viii)~~.

15. (currently amended) The method of claim 14, wherein, prior to presenting the ~~represented~~ first instance of the sequence to the user, variations are stored in a list data structure comprising a variation identification.

16. (previously presented) The method of claim 15, wherein the list data structure comprises a subsequence character that matches a “don’t care” character in a replet that matches the subsequence.

17. (currently amended) The method of claim 16, wherein, prior to presenting the ~~represented~~ first instance of the sequence to the user, the position of the subsequence character within the subsequence is stored in the list data structure.

18. (previously presented) The method of claim 15, wherein an indirection pointer points to a variation so that variations common to more than one subsequence are not stored more than once.

19. (currently amended) The method of claim 14, wherein a ~~represented~~ first instance of the sequence presented to the user is in response to a query by the user.

20. (previously presented) The method of claim 19, wherein a query specifies a replet.

21. (previously presented) The method of claim 14, wherein specifying a set of one or more replets comprises adding a new replet to a pre-existing set for which the steps of claim 14 have been performed.

22. (previously presented) The method of claim 14, wherein specifying a set of one or more replets comprises removing a replet from a pre-existing set for which the steps of claim 14 have been performed.

23. (canceled)

24. (currently amended) A computer program product for storing and presenting sequence data, the computer program product including a computer readable storage medium having instructions stored thereon for execution by a computer, wherein the instructions, when executed by the computer, cause the computer to implement a method comprising the steps of:

i) specifying a set of replets for analysis by a computer system;

ii) for each replet in the set, comparing each replet by the computer system to a sequence for determining by the computer system a subsequence of the sequence that matches each replet, if any, wherein the sequence represents a genome sequence;

iii) generating, ~~entries of match-set data structures~~ by the computer system responsive to the comparing, ~~one match-set data structure~~ respective entries of a match-set data structure, the match-set data structure having respective entries for the ~~each~~ respective replets, ~~a match-set data structure~~ each entry comprising a sequence identification to identify a ~~sub~~sequence where a match occurs and offset information to determine a position within the sequence where the matching subsequence of the sequence is located, wherein the offset information comprises a first and second position parameter,~~ the first position parameter denoting a location in the sequence and the second position parameter denoting an offset from the location~~;

iv) storing the generated entries of the match-set data structures in a computer readable memory by the computer system;

v) deleting by the computer system each matching subsequence from the sequence where it is found;

vi) concatenating by the computer system, in order, unmatched regions of sequences that remain after deleting each matching subsequence to form and store a backbone sequence;

~~vii) the computer system generating a first representation of the sequence responsive to the stored backbone sequence and at least a portion of the stored match-set data and presenting the represented sequence to a user of the computer system;~~

~~wherein one of the subsequences is matched by a certain plurality of the replets and the method further includes:~~

vii) ~~viii)~~ the computer system generating or receiving a selection of at least a first one of the replets, wherein at least a portion of one of the matching subsequences is matched by a certain plurality of the replets and the received selection includes a selection of at least one of the certain plurality of replets; [,]

viii) the computer system generating a first instance of the sequence and presenting the first instance of the sequence to a user of the computer system, wherein the generating of the sequence is responsive to the stored backbone sequence and responsive to the at least one of the stored match-set data entries corresponding to the selected at least first one of the replete, ~~first representation in step vii) is responsive to the selected one of the certain plurality of replets but not responsive to any non-selected replet,~~ wherein the first position parameter of each match-set data structure entry denotes a location in the sequence and the second position parameter of each match-set data structure entry denotes an offset from the location,~~and~~ wherein the selected at least first one of the replets has a position within the sequence, and wherein after receiving the selection the computer system updates the first and second parameters of the entries in the match-set data structure ~~for any non-selected replet associated with the selected one of the replets~~, the updating being responsive to the position of the selected at least first one of the replets;

ix) the computer system generating or receiving a selection of a second one of the replets~~ that was not selected in step viii)~~; and

x) the computer system generating and presenting a second instance ~~representation~~ of the sequence to a user~~, wherein the representation in step x) is~~ responsive to the at least second one of the replets ~~indicated by the selection in step ix) instead of the one of the replets indicated by the selection in step viii)~~ and wherein the computer system performs the generating of the second instance of the sequence ~~in step x)~~ by reference to the updated first and second parameters for the second one of the replets ~~indicated by the selection in step ix), such that the computer system performs the generating in step x) using the position parameters updated in step viii)~~.

25. (previously presented) The computer program product of claim 24, wherein storing

variations is performed using a pointer.

26. (currently amended) The computer program product of claim 24, further comprising, before presenting the ~~represented~~ first instance of the sequence to a user: identifying, replets that can be used to represent multiple subsequences.

27. (currently amended) A computer system comprising:

a processor; and

a storage device connected to the processor, wherein the storage device has stored thereon a program for controlling the processor, and wherein the processor is operative with the program to execute the program for performing the steps of:

i) specifying a set of replets for analysis by a computer system;

ii) for each replet in the set, comparing each replet by the computer system to a sequence for determining by the computer system a subsequence of the sequence that matches each replet, if any, wherein the sequence represents a genome sequence;

iii) generating, ~~entries of match-set data structures~~ by the computer system responsive to the comparing, ~~one match-set data structure~~ respective entries of a match-set data structure, the match-set data structure having respective entries for the ~~each~~respective replets, ~~a match-set data structure~~each entry comprising a sequence identification to identify a subsequence where a match occurs and offset information to determine a position within the sequence where the matching subsequence of the sequence is located, wherein the offset information comprises a first and second position parameter,~~ the first position parameter denoting a location in the sequence and the second position parameter denoting an offset from the location~~;

iv) storing the generated entries of the match-set data structures in a computer readable memory by the computer system;

v) deleting by the computer system each matching subsequence from the sequence where it is found;

vi) concatenating by the computer system, in order, unmatched regions of sequences that remain after deleting each matching subsequence to form and store a backbone sequence;

~~vii) the computer system generating a first representation of the sequence responsive to the stored backbone sequence and at least a portion of the stored match-set data and presenting the represented sequence to a user of the computer system;~~

~~wherein one of the subsequences is matched by a certain plurality of the replets and the method further includes:~~

vii) ~~viii)~~ the computer system generating or receiving a selection of at least a first one of the replets, wherein at least a portion of one of the matching subsequences is matched by a certain plurality of the replets and the received selection includes a selection of at least one of the certain plurality of replets;[,]

viii) the computer system generating a first instance of the sequence and presenting the first instance of a the sequence to a user of the computer, wherein the generating of the sequence is responsive to the stored backbone sequence and responsive to the at least one of the stored match-set data entries corresponding to the selected at least first one of the replets, ~~first representation in step vii) is responsive to the selected one of the certain plurality of replets but not responsive to any non-selected replet~~,wherein the first position parameter of each match-set data structure entry denotes a location in the sequence and the second position parameter of each match-set data structure entry denotes an offset from the location, ~~and~~ wherein the selected at least first one of the replets has a position within the sequence, and wherein after receiving the selection the computer system updates the first and second parameters of the entries in the match-set data structure ~~for any non-selected replet associated with the selected one of the replets~~, the updating being responsive to the position of the selected at least first one of the replets;

ix) the computer system generating or receiving a selection of a second one of the ~~replet~~ replets~~that was not selected in step viii)~~; and

x) the computer system generating and presenting a second instance ~~representation~~ of the sequence to a user~~, wherein the representation in step x) is~~

responsive to the at least second one of the replets ~~indicated by the selection in step ix) instead of the one of the replets indicated by the selection in step viii)~~ and wherein the computer system performs the generating of the second instance of the sequence ~~in step x)~~ by reference to the updated first and second parameters for the second one of the replets ~~indicated by the selection in step ix), such that the computer system performs the generating in step x) using the position parameters updated in step viii)~~.

28. (previously presented) The computer system of claim 27, wherein storing variations is performed using a pointer.